Europäisches Patentamt	European Patent Office	Office eur péen des brevets

Bescheinigung | Certificate | Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. | Patent application No. | Demande de brevet n°

02293228.9

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

R C van Dijk

Anmeldung Nr:
Application no.: 02293228.9
Demande no:

Anmeldetag:
Date of filing: 23.12.02
Date de dépôt:

Anmelder/Applicant(s)/Demandeur(s):

ALCATEL
54, rue La Boétie
75008 Paris
FRANCE

Bezeichnung der Erfindung/Title of the invention/Titre de l'invention:
(Falls die Bezeichnung der Erfindung nicht angegeben ist, siehe Beschreibung.
If no title is shown please refer to the description.
Si aucun titre n'est indiqué se referer à la description.)

Software traffic generator/analyser

In Anspruch genommene Priorität(en) / Priority(ies) claimed /Priorité(s) revendiquée(s)
Staat/Tag/Aktenzeichen/State/Date/File no./Pays/Date/Numéro de dépôt:

Internationale Patentklassifikation/International Patent Classification/
Classification internationale des brevets:

G06F11/28

Am Anmeldetag benannte Vertragstaaten/Contracting states designated at date of filing/Etats contractants désignées lors du dépôt:

AT BE BG CH CY CZ DE DK EE ES FI FR GB GR IE IT LI LU MC NL PT SE SI SK

SOFTWARE TRAFFIC GENERATOR/ANALYSER

The invention relates to a processor program product to be run via a processor-system for generating and/or analysing traffic signals for testing at least a part of at least one integrated-circuit-environment, which integrated-circuit-environment is designed to handle traffic signals.

Such an integrated-circuit-environment for example corresponds with an Application Specific Integrated Circuit or ASIC for example for use in a switch, router, bridge, (de)multiplexer, modem etc. or for example corresponds with several circuits like for example ASICs on a Printed Circuit Board or for example corresponds with said Printed Circuit Board etc., whereby at least one circuit or at least one connection between at least two circuits is to be tested etc.

A prior art processor program product is known from *US 5,703,788*, which discloses a library of test programs, an autodetector, an autoverifier, a failure report generator, a tools archiver, and a package information logger.

The known processor program product is disadvantageous, inter alia, due to said prior art processor program product being re-usable and efficient insufficiently.

It is an object of the invention, inter alia, of providing a processor program product as defined in the preamble which is better re-usable and more efficient.

The processor program product according to the invention is characterised in that said integrated-circuit-environment comprises said processor-system, with said processor program product comprising at least one generic module and at least one specific module, with at least one specific module being designed for interfacing said computer program product with a protocol used in said integrated-circuit-environment.

By providing said processor program product with said generic module which is re-usable in next generations of said processor program product due

to being generic, the re-usability of said processor program product has been increased. By providing said processor program product with said specific module which interfaces with a protocol used in said integrated-circuit-environment, the processor program product can be adapted to the kind of integrated-circuit-environment to be tested. By using the processor-system already present in said integrated-circuit-environment, no additional hardware is required, and the efficiency is increased.

The part of the integrated-circuit-environment to be tested may correspond with at least a part of said processor-system or with at least a part of an other circuit forming part of said integrated-circuit-environment or with at least one connection between at least two circuits (including said processor-system or not).

Of course firstly, when testing the same kind of integrated-circuit-environment, the same specific module can also be re-used. Of course secondly, re-usability will generally increase the overall efficiency.

A first embodiment of the processor program product according to the invention is defined by claim 2.

By using a processor-system comprising at least one host processor, generated traffic signals firstly flow from said host processor to a buffer and secondly flow from said buffer to at least one further circuit of said integrated-circuit-environment, and an efficient procedure for testing the integrated-circuit-environment by using its own host processor in combination with the processor program product according to the invention has been created, and the additional hardware-problem of how to get the generated traffic signals from the integrated-circuit-environment's own host processor into other parts of said integrated-circuit-environment has been solved.

Said protocol used in said integrated-circuit-environment for example corresponds with a protocol used in said host processor or with a protocol used in one or more of the other circuits etc.

A second embodiment of the processor program product according to the invention is defined by claim 3.

By letting generated traffic signals leave said processor program product via a software traffic sender, with traffic signals to be analysed arriving at said processor program product via a software traffic receiver, the additional interface-problem of how to get the generated traffic signals from the software processor program product into the hardware parts of said integrated-circuit-environment has been solved.

A third embodiment of the processor program product according to the invention is defined by claim 4.

By designing said specific module for interfacing a traffic protocol, integrated-circuit-environments based upon different traffic protocols can be tested by using the same generic module in combination with different specific modules.

A fourth embodiment of the processor program product according to the invention is defined by claim 5.

By designing said specific module for interfacing an Internet-Protocol or an Asynchronous-Transfer-Mode-Protocol or an Ethernet-Protocol, integrated-circuit-environments designed for an Internet environment or an Asynchronous-Transfer-Mode environment or an Ethernet environment can be tested. Of course, further environments are not to be excluded.

A fifth embodiment of the processor program product according to the invention is defined by claim 6.

By designing said specific module for interfacing a bus protocol, integrated-circuit-environments based upon different bus protocols can be tested by using the same generic module in combination with different specific modules.

A sixth embodiment of the processor program product according to the invention is defined by claim 7.

By designing said specific module for interfacing a flexbus4 protocol or a SPI4.2 protocol (System Physical Interface), integrated-circuit-environments having a flexbus4 or a SPI4.2 can be tested. Of course, further protocols are not to be excluded.

Preferably at least one generic module is designed to operate in dependence of adjustable parameters, for adjusting the test environment. In case of said adjustable parameters comprising a bandwidth parameter, the bandwidth of the traffic signals in the test can be adjusted. In case of said adjustable parameters comprising a flow parameter, the number of flows of the traffic signals in the test can be adjusted. Of course, further adjustable parameters are not to be excluded.

The invention further relates to a processor-system for running a processor program product for generating and/or analysing traffic signals for testing at least a part of at least one integrated-circuit-environment, which integrated-circuit-environment is designed to handle traffic signals.

The processor-system according to the invention is characterised in that said integrated-circuit-environment comprises said processor-system, with said processor program product comprising at least one generic module and at least one specific module, with at least one specific module being designed for interfacing said computer program product with a protocol used in said integrated-circuit-environment.

The invention yet further relates to an integrated-circuit-environment to be tested via a processor-system for running a processor program product for generating and/or analysing traffic signals for testing at least a part of said integrated-circuit-environment, which integrated-circuit-environment is designed to handle traffic signals.

The integrated-circuit-environment according to the invention is characterised in that said integrated-circuit-environment comprises said processor-system, with said processor program product comprising at least one generic module and at least one specific module, with at least one specific module being designed for interfacing said computer program product with a protocol used in said integrated-circuit-environment.

The invention also relates to a method for generating and/or analysing traffic signals via a processor-system for testing at least a part of at least one integrated-circuit-environment designed to handle traffic signals.

The method according to the invention is characterised in that said integrated-circuit-environment comprises said processor-system, with said method comprising at least one generic step and at least one specific step, with at least one specific step being performed for interfacing with a protocol used in said integrated-circuit-environment.

Embodiments of the processor-system according to the invention and of the integrated-circuit-environment according to the invention and of the method according to the invention correspond with the embodiments of the processor program product according to the invention.

The invention is based upon an insight, inter alia, that large parts of processor program products for generating and/or analysing traffic signals for testing at least parts of integrated-circuit-environments should be re-usable, and is based upon a basic idea, inter alia, that this re-usability can be increased by creating generic modules and specific modules.

The invention solves the problem, inter alia, of providing a processor program product having a better re-usability, and is advantageous, inter alia, in that the use of the processor-system of said integrated-circuit-environment avoids the need for additional hardware, which increases the efficiency.

These and other aspects of the invention will be apparent from and elucidated with reference to the embodiments(s) described hereinafter.

Figure 1 illustrates in block diagram form an overview of the processor program product according to the invention,

Figure 2 illustrates in block diagram form an integrated-circuit-environment according to the invention,

Figure 3 illustrates in block diagram form a more detailed view of the processor program product according to the invention for generating traffic signals, and

Figure 4 illustrates in block diagram form a more detailed view of the processor program product according to the invention for analysing traffic signals.

The overview of the processor program product according to the invention shown in Figure 1 comprises a processor program product 1 comprising a software traffic generator 2,4 and a software traffic analyser 3,5. Software traffic generator 2,4 comprises a generic module 2 and a specific module 4, and software traffic analyser 3,5 comprises a generic module 3 and a specific module 5. Generic module 2 receives information from a user module 9 and sends information to a software traffic sender 6. Generic module 3 receives information from a software traffic receiver 7 and sends information to user-module 9. Between said traffic sender 6 and traffic receiver 7, an integrated-circuit-environment 8 is situated. Although just one specific module 4,5 is shown in the drawing, more than one may be present, for example coupled serially or parallelly to each other and/or to the corresponding generic module 2,3.

The integrated-circuit-environment 8 according to the invention shown in Figure 2 comprises a host processor 80 with a memory 89. An in/output of host processor 80 is coupled via a PCI bus to an in/output of a first circuit 81 communicating with a buffer 87 and to an in/output of a second circuit 82 communicating with a buffer 88. A further in/output of a first circuit 81 is coupled to an in/output of a further circuit 86 and a yet further in/output of first circuit 81 is coupled to an in/output of a further circuit 83 of which a further in/output is coupled to a further in/output of second circuit 82. An other in/output of further circuit 83 is coupled to an in/output of a further circuit 84 and a yet other in/output of further circuit 83 is coupled to an in/output of a further circuit 85, of which a further in/output is coupled to a further in/output of a further circuit 84. Further circuit 85 is further coupled to host processor 80.

The processor program product 1 runs via processor-system 80 (more particularly via host processor 80) for generating (via software traffic generator 2,4) and/or analysing (via software traffic analyser 3,5) traffic signals for testing integrated-circuit-environment 8 comprising said processor-system 80. Such an integrated-circuit-environment 8 for example corresponds

with an Application Specific Integrated Circuit or ASIC for example for use in a switch, router, bridge, (de)multiplexer, modem etc. or for example corresponds with several circuits like for example ASICs on a Printed Circuit Board or for example corresponds with said Printed Circuit Board etc., whereby at least one circuit or at least one connection between at least two circuits is to be tested etc.

By providing said processor program product 1 with said generic module 2,3 which is re-usable in next generations of said processor program product 1 due to being generic, the re-usability of said processor program product 1 has been increased. By providing said processor program product 1 with said specific module 4,5 which interfaces with a protocol used in said integrated-circuit-environment 8, the processor program product 1 can be adapted to the kind of integrated-circuit-environment 8 to be tested. By using the processor-system 80 already present in said integrated-circuit-environment 8, no additional hardware is required. Said protocol used in said integrated-circuit-environment 8 for example corresponds with a protocol used in said host processor 80 or with a protocol used in one or more of the further circuits 83,84,85 or with a protocol used in the first or second circuit 81,82 etc.

The part of the integrated-circuit-environment 8 to be tested may correspond with at least a part of said processor-system 80 or with at least a part of an other circuit 81-88 forming part of said integrated-circuit-environment 8 or with at least one connection between at least two circuits 80-88 (including said processor-system 80 or not).

Said specific modules 4,5 interface traffic protocols like Internet-Protocols or Asynchronous-Transfer-Mode-Protocols or Ethernet-Protocols and/or interface bus protocols like flexbus4 protocols or SPI4.2 (System Physical Interface) protocols. But other protocols are not to be excluded.

Said generic modules 2,3 operate in dependence of adjustable parameters for adjusting the test environment. These adjustable parameters comprise a bandwidth parameter for adjusting the bandwidth of the traffic signals in the test and/or comprise a flow parameter for adjusting the number

of flows of the traffic signals in the test. Other adjustable parameters are not to be excluded.

Said traffic sender 6 forms a tool for interfacing the processor program product 1 according to the invention (more particularly modules 2,4) with the hardware. Similarly, said traffic receiver 7 forms a tool for interfacing the processor program product 1 according to the invention (more particularly modules 3,5) with the hardware.

Generated traffic signals firstly flow from said host processor 80 via said first circuit 81 to said buffer 87 and are stored in buffer 87. Secondly they flow from said buffer 87 to at least one further circuit 83,84,85 of said integrated-circuit-environment 8, and an efficient procedure for testing the integrated-circuit-environment 8 by using its own host processor 80 in combination with the processor program product 1 according to the invention has been created, and the additional hardware-problem of how to get the generated traffic signals from the integrated-circuit-environment's 8 own host processor 80 into other parts of said integrated-circuit-environment 8 has been solved.

Via further circuits 83, 84 and 85 the traffic signals return via second circuit 82 in buffer 88 where they are stored. Then they are supplied to host processor 80 for being analysed etc.

Said generated traffic signals leave said processor program product 1 via a software traffic sender 6, with traffic signals to be analysed arriving at said processor program product 1 via a software traffic receiver 7, to solve the additional interface-problem of how to get the generated traffic signals from the software processor program product 1 into the hardware parts of said integrated-circuit-environment 8 and vice versa.

The more detailed view of the processor program product 1 according to the invention for generating traffic signals as shown in figure 3 comprises a processor program product 2+4 comprising specific module 4 and generic modules 20 and 21, with generic module 20 receiving information from user module 9 and sending information to generic module 21 and to specific module 4 and to traffic sender 6, which sends traffic signals to integrated-

circuit-environment 8. Generic module 21 and specific module 4 also receive information from user module 9 and send back information to generic module 20.

Generic module 20 for example builds packet data units and transmits the builded packet data units, generic module 21 for example constructs a payload (like for example all zero, all one, increasing bytes, decreasing bytes etc.) and specific module 4 for example constructs headers (like for example in accordance with Internet Protocol version 4 or 6 etc.).

The following functions can for example be found in the generating part of the processor program product:

- non-functional interface functions (like for example an initialisation function, a setting function for setting the seed used for PRB generation (Pseudo Random Bit Pattern), a closing function etc.),
- packet data unit definition interface functions (like for example a header definition function for getting a header size (in bytes) based on the header definition, a header construction function for constructing a packet data unit header (the necessary memory should have been allocated by the caller of this function, the header can depend on the payload (e.g. size field in Internet Protocol version 4 or 6), therefore it can be necessary to have the entire packet data unit definition to build the header), a payload construction function, a packet data unit construction function, a packet data unit destruction function etc.),
- traffic definition interface functions (like for example a creating function for creating traffic according to a per flow definition (the algorithm interleaves the packet data units of each flow, the interleaving will be better when the weights of the packet data units are smaller), a transmitting function for transmitting existing packet data units, a sending function for sending existing packet data units to the traffic sender etc.).

The more detailed view of the processor program product 1 according to the invention for analysing traffic signals as shown in Figure 4 comprises a processor program product 3+5 comprising a generic module 30 and a buffer module 31. Generic module 30 receives information from user module

9 and sends information to buffer module 31, which receives information from packet receiver 7 which receives traffic signals to be analysed from integrated-circuit-environment 8. Buffer module 31 further transmits information to a pre-process module 10 for example forming part of user module 9.

The following functions can for example be found in the analysing part of the processor program product:

- non-functional interface functions (like for example an opening/starting function for opening/starting the receiving mechanism, a stopping function for stopping the receiving mechanism, a request function (the traffic receiver asks for a buffer to store a received packet data unit), a result function (the traffic receiver has written a packet data unit), an indication function for indicating whether the analysing part is able to receive information, an overflow function for indicating whether a buffer is overflown etc.),
- functional interface functions (like for example a running function for running an analyse cascade over all buffered packet data units, a storing function (stubbing pre-process block: this function stores the packet data unit, it is to be used only in a pre-process cascade (not in an analyse cascade) and should terminate the pre-process cascade), a dropping function (stubbing pre-process block: this function drops the packet data unit, it is to be used only in a pre-process cascade (not in an analyse cascade) and should terminate the pre-process cascade), a stopping function (pre-process block – stops the packet data unit reception mechanism (not the cascade), this function is to be used only in a pre-process cascade (not in an analyse cascade)), processing functions (general processing block – counts the packet data units that pass, the user should set the counter to its initial value (zero), the counter saturates at its maximum value + general processing block – switches when counter reached x + general processing block – switches when packet data unit length is greater then x), a returning function (stubbing block, this function is doing nothing but returning, it can be used to terminate an unused branch in a block).

Generally, the processor program product operates from a host platform. This means that the traffic is generated and analysed in software on

a host processor. In this way it is possible to make the core of the processor program product independent of the underlying hardware. To use this processor program product on a specific hardware configuration it is necessary to port it to the underlying hardware. This porting is done via the traffic sender and the traffic receiver (porting layer). The generating part and the analysing part of the processor program product are in principle independent and they can run on the same or on separated processors at the same time. The parts are not multi-thread proof: only one instance of each part can run on a processor. This means that the analysing part needs to be interrupt-based, as the analysing part will have to accept packets while the generating part is still sending them (if they are running on the same processor). But of course, said parts can be made multi-thread proof.

The expression "for" in for example "for generating" and "for analysing" and "for testing" and "for interfacing" etc. does not exclude that other functions are performed as well, simultaneously or not. The expressions "X coupled to Y" and "a coupling between X and Y" and "coupling/couples X and Y" etc. do not exclude that an element Z is in between X and Y. The expressions "P comprises Q" and "P comprising Q" etc. do not exclude that an element R is comprises/included as well. The terms "a" and "an" do not exclude the possible presence of one or more pluralities. A processor-system generally comprises one or more processors.

The invention is based upon an insight, inter alia, that large parts of processor program products 1 for generating and/or analysing traffic signals for testing at least parts of integrated-circuit-environments 8 should be re-usable, and is based upon a basic idea, inter alia, that this re-usability can be increased by creating generic modules 2,3 and specific modules 4,5.

The invention solves the problem, inter alia, of providing a processor program product 1 having a better re-usability, and is advantageous, inter alia, in that the use of the processor-system 80 of said integrated-circuit-environment 8 avoids the need for additional hardware, which increases the efficiency.

CLAIMS

1. Processor program product (1) to be run via a processor-system (80) for generating and/or analysing traffic signals for testing at least a part of at least one integrated-circuit-environment (8), which integrated-circuit-environment (8) is designed to handle traffic signals,

characterised in that said integrated-circuit-environment (8) comprises said processor-system (80), with said processor program product (1) comprising at least one generic module (2,3) and at least one specific module (4,5), with at least one specific module (4,5) being designed for interfacing said computer program product (1) with a protocol used in said integrated-circuit-environment (8).

2. Processor program product (1) according to claim 1,

characterised in that said processor-system (80) comprises at least one host processor (80), with generated traffic signals flowing from said host processor (80) to a buffer (87) and from said buffer (87) to at least one further circuit (83,84,85) of said integrated-circuit-environment (8).

3. Processor program product (1) according to claim 2,

characterised in that generated traffic signals leave said processor program product (1) via a software traffic sender (6), with traffic signals to be analysed arriving at said processor program product (1) via a software traffic receiver (7).

4. Processor program product (1) according to claim 1, 2 or 3,

characterised in that said protocol comprises a traffic protocol.

5. Processor program product (1) according to claim 4,

characterised in that said traffic protocol comprises an Internet-Protocol or an Asynchronous-Transfer-Mode-Protocol or an Ethernet-protocol.

6. Processor program product (1) according to claim 1, 2, 3, 4 or 5,

characterised in that said protocol comprises a bus protocol.

7. Processor program product (1) according to claim 6,

characterised in that said bus protocol comprises a flexbus4 protocol or a SPI4.2 protocol.

8. Processor-system (80) for running a processor program product (1) for generating and/or analysing traffic signals for testing at least a part of at least one integrated-circuit-environment (8), which integrated-circuit-environment (8) is designed to handle traffic signals,

characterised in that said integrated-circuit-environment (8) comprises said processor-system (80), with said processor program product (1) comprising at least one generic module (2,3) and at least one specific module (4,5), with at least one specific module (4,5) being designed for interfacing said computer program product (1) with a protocol used in said integrated-circuit-environment (8).

9. Integrated-circuit-environment (8) to be tested via a processor-system (80) for running a processor program product (1) for generating and/or analysing traffic signals for testing at least a part of said integrated-circuit-environment (8), which integrated-circuit-environment (8) is designed to handle traffic signals,

characterised in that said integrated-circuit-environment (8) comprises said processor-system (80), with said processor program product (1) comprising at least one generic module (2,3) and at least one specific module (4,5), with at least one specific module (4,5) being designed for interfacing said computer program product (1) with a protocol used in said integrated-circuit-environment (8).

10. Method for generating and/or analysing traffic signals via a processor-system (80) for testing at least a part of at least one integrated-circuit-environment (8) designed to handle traffic signals,

characterised in that said integrated-circuit-environment (8) comprises said processor-system (80), with said method comprising at least one generic step and at least one specific step, with at least one specific step being performed for interfacing with a protocol used in said integrated-circuit-environment (8).

ABSTRACT

Processor program products (1) to be run via a processor-system (80) for generating and/or analysing traffic signals for testing at least parts of integrated-circuit-environments (8) designed to handle traffic signals are provided with generic modules (2,3) and specific module (4,5) to increase the re-usability. Said specific modules (4,5) are designed for interfacing said computer program product (1) with a protocol used in said integrated-circuit-environment (8), like a traffic protocol like an Internet-Protocol or an Asynchronous-Transfer-Mode-Protocol or an Ethernet-Protocol, or like a bus protocol like a flexbus4 protocol or a SPI4.2 protocol. Thereby, the processor-system (80) already present in said integrated-circuit-environment (8) and comprising a host processor (80) is used for running said processor program product (1), which saves additional hardware, with generated traffic signals flowing from said host processor (80) to a buffer (87) and from said buffer (87) to at least one further circuit (83,84,85) of said integrated-circuit-environment (8).

Figure 1

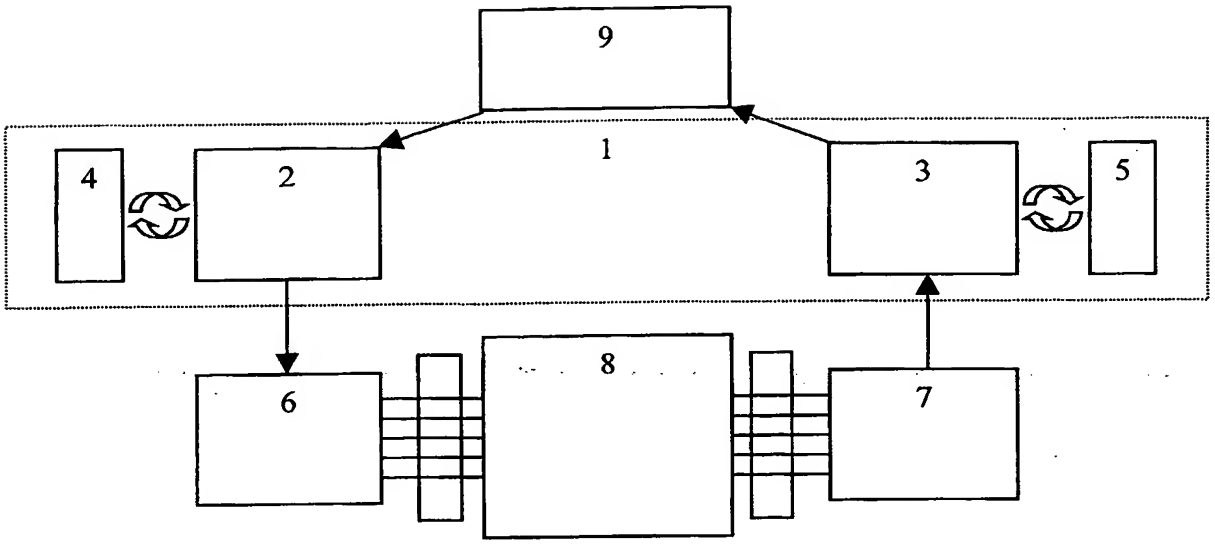

Fig. 1

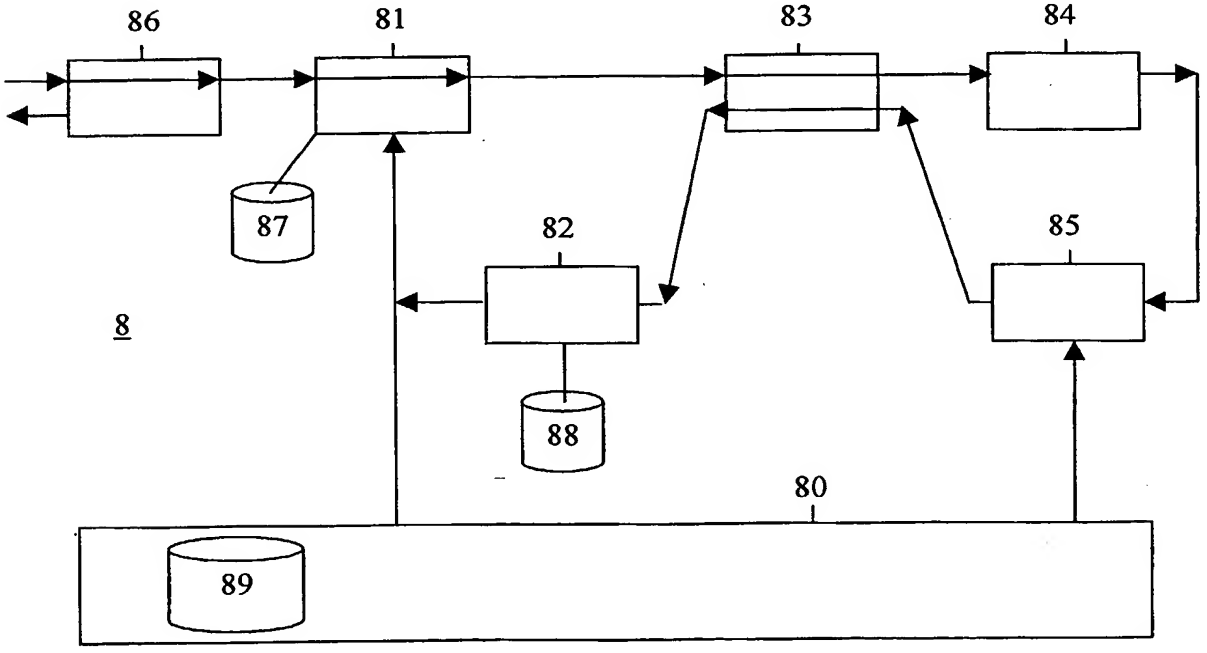

Fig. 2

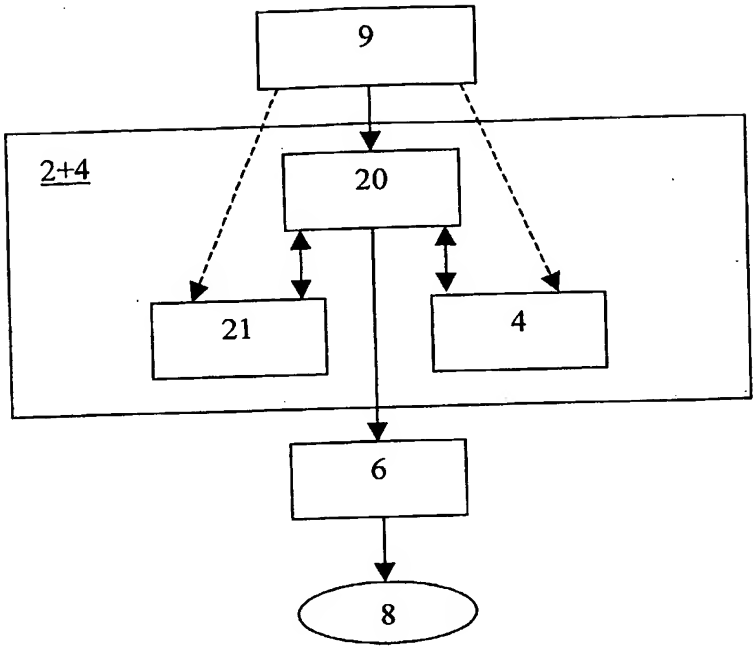

Fig. 3

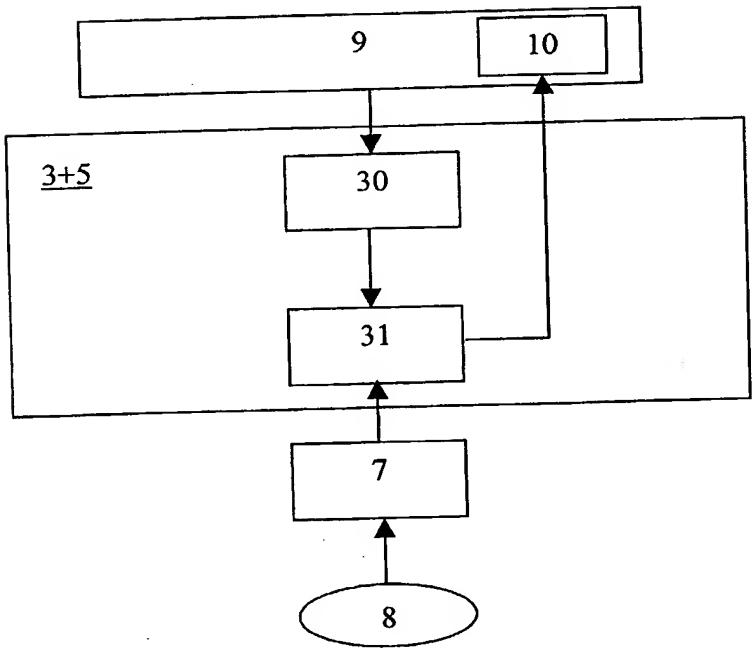

Fig. 4